Model-based clustering for high-dimension data. Application to functional data.

Emilie Devijver

MODEL-BASED CLUSTERING FOR HIGH-DIMENSIONAL DATA. APPLICATION TO FUNCTIONAL DATA.

EMILIE DEVIJVER

Abstract. Finite mixture regression models are useful for modeling the relationship between response and predictors, arising from different subpopulations. In this article, we study high-dimensional predictors and high-dimensional response, and propose two procedures to deal with this issue. We propose to use the Lasso estimator to take into account the sparsity, and a penalty on the rank, to take into account the matrix structure. Then, we extend these procedures to the functional case, where predictors and responses are functions. For this purpose, we use a wavelet-based approach. Finally, for each situation, we provide algorithms, and apply and evaluate our methods both on simulations and real datasets.

1. Introduction

Owing to the increase of high-dimensional datasets, regression models for multivariate response and high-dimensional predictors have become important tools.
The goal of this article is to describe two procedures which cluster data. We focus on the model-based clustering. Each cluster is represented by a parametric conditional distribution, the entire dataset being modeled by a mixture of these distributions. This provides a rigorous statistical framework, and allows to understand the role of each variable in the clustering process. The model considered is then, if $(Y_i, X_i) \in \mathbb{R}^m \times \mathbb{R}^p$ belongs to the component r, there exists an unknown $p \times m$ matrix of coefficient β_r such that

$$Y_i = \beta_r X_i + E. \tag{1}$$

We will work with high-dimensional datasets, that is to say $p \times m$ could be larger than the sample size n, then we have to reduce the dimension. Two ways will be considered here, coefficients sparsity and ranks sparsity. Let describe them into a linear model.
We could work with a sparse model if the matrix β could be estimated by a matrix with few nonzero coefficients. The well-known Lasso estimator, introduced by Tibshirani in 1996 ([22]), is the solution chosen here. Indeed, the Lasso estimator is used for variable selection, cite for example Meinshausen and Bühlmann in [14] for stability selection results. We could also cite the book of Bühlmann and van de Geer ([4]) for an overview of the Lasso estimator.
If we look for rank sparsity for β, we have to assume that a lot of regressors are linearly dependent. This approach date's back to the 1950's, and was initiated by Anderson ([1]). Izenman ([11]) introduced the term of reduced-rank regression for this class of models. A number of important works followed, cite for example Giraud ([10]) and Bunea ([5]) for recent articles.
Nevertheless, this linear regression model is appropriate for modeling the relationship between response and predictors when the reliance is the same for all observations, and it is inadequate for settings in which the coefficient function differs across subgroups of the observations, we will consider mixture models. Moreover, we could perform estimations with clustering data which are more similar.
In this paper, functional datasets (functional predictors and functional response) will be considered. They have been studied in the book of Ramsay and Silverman ([19]). With the increase of functional datasets, a lot of recent works have been done on regression models: for example, cite the article of Ciarleglio ([6]) which deals with scalar response and functional regressors. In our case, wavelet basis will be considered. We could apply the model (1) on the wavelet coefficients of the datasets. Wavelets are particularly well suited to handle many types of functional data. Indeed, they represent global and local attributes of functions, and can deal with discontinuities. To deal with the sparsity previously mentioned, a large class of functions can be well represented with few non-zero coefficients, for any suitable wavelet.
We propose here two procedures which cluster high-dimensional data or data described by a functional variable, explained by high-dimensional predictors or by predictor variables arising from sampling continuous curves. Note that we estimate the number of components, parameters of each model, and the

Date: September 3, 2014.
Key words and phrases. Model-based clustering, regression, high-dimension, functional data.

proportions. We assume we do not have any knowledge about the model, except that it could be well approximated by sparse Gaussian regression model. The high dimensional problem is solved by using variable selection to detect relevant variables. Since the structure of interest may often be contained into a subset of the available variables and many attributes may be useless or even harmful to detect a reasonable clustering structure, it is important to select the relevant clustering variables. Moreover, removing irrelevant variables enables to get simpler modeling and can largely enhance interpretability. Nevertheless, in our procedures, instead of focus of the regularization parameter (which is a difficult choice), we construct a model collection for different values of the regularization parameter.
All of these procedures are mainly based on three recent works. Firstly, we could cite the article of Städler et al. [21], which studies finite mixture regression models. Even if we work on a multivariate version of these models, the approach considered in the article [21] is adopted here. The second, Meynet and Maugis article [15], deals with model-based clustering in density estimation. They give a procedure, called Lasso-MLE procedure, which determines the number of clusters, the set of relevant variables for the clustering, and a clustering of the observations, with high-dimensional data. We extend this procedure with conditional densities. Finally, we could cite an article of Giraud [10]. It suggests a low-rank estimator for the linear model. To take into account the matrix structure, we will consider this approach for estimation in our mixture models.
In this paper, the Lasso-MLE procedure is generalized to the regression model, and develop it in a multivariate framework. We consider finite mixture of Gaussian regression model. We propose two different procedures, considering more or less the matrix structure. Both of them have the same frame. Firstly, an ℓ_1-penalized likelihood approach is considered to determine potential sets of relevant variables. Introduced by Tibshirani in [22], the Lasso is used to select variables. This allows one to efficiently construct a data-driven model subcollection with reasonable complexity, even for high-dimensional situations, with different sparsities. The second step of the procedures is to estimate parameters in a better way than by the Lasso. Then, we select a model among the collection using the slope heuristic, developed by Birgé and Massart in [3]. Differences between the both procedures are the estimation of parameters in each model. The first one, later called Lasso-MLE procedure, uses the maximum likelihood estimator rather than the ℓ_1-penalized maximum likelihood estimator. It avoids estimation problems due to the ℓ_1-penalization shrinkage. The second one, called Lasso-rank procedure, deals with low rank estimation. For each model in the collection, we construct a subcollection of models with conditional means estimated by various low ranks matrices. It leads to sparsity and for the coefficients, and for the rank.
The paper is organized as follows. Section 2 deals with Gaussian mixture regression models. It describes the model collection that we will consider. In Section 3, we describe both procedures that we propose to solve the problem of clustering high-dimensional regression data. Section 4 presents an illustrative example, to highlight each choice involved by both procedures. Section 5 states the functional data case, with a description of the projection proposed to convert these functions into coefficients data. We end this section by simulations, on simulated and on benchmark data. Finally, a conclusion section ends this paper.

2. Gaussian mixture regression models

We have to construct a statistical framework on the observations. Because we estimate the conditional densities by multivariate Gaussian in each cluster, then the model used is a finite Gaussian mixture regression model. Städler et al ([21]) describe this model, when X is multidimensional, and Y is scalar. We generalize it in the multivariate case in this section. Moreover, we will construct a model collection of Gaussian mixture regression models, with several sparsities.

2.1. Gaussian mixture regression.

We observe n independent couples $((x_i, y_i))_{1\leq i\leq n}$ of variables (X, Y), with $Y \in \mathbb{R}^m$ and $X \in \mathbb{R}^p$ coming from a probability distribution with unknown conditional density, denoted by s. We want to perform model-based clustering, then we assume that data come from a mixture density: $s(y|x) = \sum_{r=1}^{k} \pi_r s_r(y|x)$. To get a Gaussian mixture regression model, we suppose that, if $Y|X$ belongs to the cluster r,

$$Y = \beta_r X + \epsilon$$

where $\epsilon \sim N(0, \Sigma_r)$. We then assume that s_r is a multivariate Gaussian density.
Thus, the random response variable $Y \in \mathbb{R}^m$ depends on a set of explanatory variables, written $X \in \mathbb{R}^p$, through a regression-type model. Some assumptions are in order.

- The variables $Y_i|X_i$ are independent, for all $i = 1, \ldots, n$;

- we have $Y_i|X_i = x_i \sim s_\xi(y|x_i)dy$, with

$$s_\xi(y|x) = \sum_{r=1}^{k} \frac{\pi_r}{(2\pi)^{\frac{m}{2}} \det(\Sigma_r)^{1/2}} \exp\left(-\frac{(y-\beta_r x)^t \Sigma_r^{-1}(y-\beta_r x)}{2}\right)$$
$$\xi = (\pi_1, \ldots, \pi_k, \beta_1, \ldots, \beta_k, \Sigma_1, \ldots, \Sigma_k) \in \left(\Pi_k \times (\mathbb{R}^{p\times m})^k \times (\mathbb{S}_{++}^m)^k\right)$$
$$\Pi_k = \left\{(\pi_1, \ldots, \pi_k); \pi_r > 0 \text{ for } r \in \{1, \ldots, k\} \text{ and } \sum_{r=1}^{k} \pi_r = 1\right\}$$
$\mathbb{S}_{++}^m$ is the set of symmetric positive definite matrices on $\mathbb{R}^m$.

Then, we want to estimate the conditional density function s_ξ from the observations. For all $r \in \{1, \ldots, k\}, \beta_r$ is the matrix of regression coefficients, and Σ_r is the covariance matrix in the mixture component r. The π_rs are the mixture proportions. Actually, for all $r \in \{1, \ldots, k\}$, for all $z \in \{1, \ldots, m\}$, $\beta_{r,z}^t x = \sum_{j=1}^{p} \beta_{r,j,z} x_j$ is the zth component of the mean of the mixture component r for the conditional density $s_\xi(.|x)$.
In order to have a scale-invariant maximum likelihood estimator, and to have a convex optimization problem, we reparametrize the model described above by generalizing the reparametrization described in [21].
We then define $\Phi_r = P_r \beta_r$, in which ${}^t P_r P_r = \Sigma_r^{-1}$ (this is the Cholesky decomposition of the positive definite matrix Σ_r^{-1}). Our hypotheses could now be rewritten:

- the variables $Y_i|X_i$ are independent, for $i = 1 \ldots n$;
- the variables $Y_i|X_i = x_i \sim h_\theta(y|x_i)dy$, for $i = 1 \ldots n$, with

$$h_\theta(y|x) = \sum_{r=1}^{k} \frac{\pi_r \det(P_r)}{(2\pi)^{m/2}} \exp\left(-\frac{(P_r y - \Phi_r x)^t (P_r y - \Phi_r x)}{2}\right)$$
$$\theta = (\pi_1, \ldots, \pi_k, \Phi_1, \ldots, \Phi_k, P_1, \ldots, P_k) \in \left(\Pi_k \times (\mathbb{R}^{p\times m})^k \times T_m^k\right)$$
$$\Pi_k = \left\{(\pi_1, \ldots, \pi_k); \pi_r > 0 \text{ for } r \in \{1, \ldots, k\} \text{ and } \sum_{r=1}^{k} \pi_r = 1\right\}$$
T_m is the set of lower triangular matrix with non-negative diagonal entries.

The log-likelihood of this model is equal to

$$l(\theta) = \sum_{i=1}^{n} \log\left(\sum_{r=1}^{k} \frac{\pi_r \det(P_r)}{(2\pi)^{m/2}} \exp\left(-\frac{(P_r y_i - \Phi_r x_i)^t (P_r y_i - \Phi_r x_i)}{2}\right)\right);$$

and the maximum log-likelihood estimator (later denoted by MLE) is

$$\hat{\theta} := \underset{\theta\in\Theta}{\operatorname{argmin}} \left\{-\frac{1}{n} l(\theta)\right\}.$$

This estimator is scale-invariant, and the optimization is convex in each cluster.
Since we deal with the $p \times m >> n$ case, this estimator has to be regularized to obtain accurate estimates. As a result, we propose the ℓ_1-norm penalized MLE

$$\hat{\theta}_\lambda := \underset{\theta\in\Theta}{\operatorname{argmin}} \left\{-\frac{1}{n} l_\lambda(\theta)\right\};$$

where

$$-\frac{1}{n} l_\lambda(\theta) = -\frac{1}{n} l(\theta) + \lambda \sum_{r=1}^{k} \pi_r ||\Phi_r||_1;$$

where $||\Phi_r||_1 = \sum_{j=1}^{p} \sum_{z=1}^{m} |\Phi_{r,j,z}|$, and with λ to specify. This estimator is not the usual ℓ_1-estimator, called the Lasso, introduced by Tibshirani in [22]. It penalizes the ℓ_1-norm of the coefficients and small variances simultaneously, which has some close relations to the Bayesian Lasso (Park and Casella [18]). Moreover, the reparametrization allows us to consider non-standardized data.

Notice that we restrict ourselves in this paper to diagonal covariance matrices which are dependent of the clusters, that is to say for all $r \in \{1, \dots, k\}$, $\Sigma_r = \begin{pmatrix} \sigma_{r,1}^2 & \dots & 0 \\ 0 & \ddots & 0 \\ 0 & \dots & \sigma_{r,m}^2 \end{pmatrix}$. Then, with the renormalization described above, the restriction becomes, for all $r \in \{1, \dots, k\}$, $P_r = \begin{pmatrix} \rho_{r,1} & \dots & 0 \\ 0 & \ddots & 0 \\ 0 & \dots & \rho_{r,m} \end{pmatrix}$.

2.2. **EM algorithm.** From an algorithmic point of view, we will use an EM algorithm to compute the MLE and the ℓ_1-norm penalized MLE. The EM algorithm was introduced by Dempster et al. in [7] to approximate the maximum likelihood estimator of parameters of mixture model. This is an iterative process based on the minimization of the expectation of the empirical contrast for the complete data conditionally to the observations and the current estimation of the parameter $\theta^{(\text{ite})}$ at each iteration (ite). Thanks to the Karush-Kuhn-Tucker conditions, we could extend the second step to compute our maximum likelihood estimators, penalized or not, as it was done in the scalar case in [21]. We therefore obtain for updating formulae:

$$\pi_r^{(\text{ite}+1)} = \pi_r^{(\text{ite})} + t^{(\text{ite})} \left(\frac{\sum_{i=1}^n \hat{\gamma}_{i,r}}{n} - \pi_r^{(\text{ite})} \right); \tag{2}$$

$$\rho_{r,z}^{(\text{ite}+1)} = \frac{n_r \langle \tilde{y}_{.,z,r}, \Phi_{r,z}^{(\text{ite})} \tilde{x}_{.,r} \rangle + \sqrt{\Delta}}{2 n_r ||\tilde{y}_{.,z,r}||_2^2}; \tag{3}$$

$$\Phi_{r,j,z}^{(\text{ite}+1)} = \begin{cases} \frac{-S_{r,j,z}^{(\text{ite})} + n\lambda(\pi_r^{(\text{ite})})}{||\tilde{x}_{.,r,j}||_2^2} & \text{if } S_{r,j,z}^{(\text{ite})} > n\lambda(\pi_r^{(\text{ite})}); \\ -\frac{S_{r,j,z}^{(\text{ite})} + n\lambda(\pi_r^{(\text{ite})})}{||\tilde{x}_{.,r,j}||_2^2} & \text{if } S_{r,j,z}^{(\text{ite})} < -n\lambda(\pi_r^{(\text{ite})}); \\ 0 & \text{else ;} \end{cases} \tag{4}$$

with, for $j \in \{1, \dots, p\}, r \in \{1, \dots, k\}, z \in \{1, \dots, m\}$,

$$S_{r,j,z}^{(\text{ite})} = -\sum_{i=1}^n \tilde{x}_{i,r,j} \rho_z^{(\text{ite})} \tilde{y}_{i,r,z} + \sum_{j_2=1, j_2 \neq j}^p \tilde{x}_{i,r,j} \tilde{x}_{i,r,j_2} \Phi_{r,j_2,z}^{(\text{ite})}; \tag{5}$$

$$n_r = \sum_{i=1}^n \hat{\gamma}_{i,r};$$

$$(\tilde{y}_{i,r}, \tilde{x}_{i,r}) = \sqrt{\hat{\gamma}_{i,r}} (y_i, x_i);$$

$$\Delta = \left(-n_r \langle \tilde{y}_{.,z,r}, \Phi_{r,z} \tilde{x}_{.,r} \rangle\right)^2 - 4 ||\tilde{y}_{.,z,r}||_2^2;$$

$$\hat{\gamma}_{i,r} = \frac{\pi_r^{(\text{ite})} \left(\prod_{z=1}^m \rho_{r,z}^{(\text{ite})} \right) \exp\left(-\frac{1}{2} \left(P^{(\text{ite})} y_i - \Phi_r^{(\text{ite})} x_i \right)^t \left(P^{(\text{ite})} y_i - \Phi_r^{(\text{ite})} x_i \right) \right)}{\sum_{l=1}^k \pi_l^{(\text{ite})} \left(\prod_{z=1}^m \rho_{r,z}^{(\text{ite})} \right) \exp\left(-\frac{1}{2} \left(P^{(\text{ite})} y_i - \Phi_l^{(\text{ite})} x_i \right)^t \left(P^{(\text{ite})} y_i - \Phi_l^{(\text{ite})} x_i \right) \right)}; \tag{6}$$

and $t^{(ite)} \in (0, 1]$, the largest value in the grid $\{\delta^k, k \in \mathbb{N}\}$, $0 < \delta < 1$, such that the function is not increasing.
See Appendix 8.1 for more details.
In our case, the EM algorithm corresponds to switch between the E-step which corresponds to the calculus of (2), (3) and (4), and the M-step, which corresponds to the calculus of (6).
We need to precise the initialization and the stopping rules. Indeed, we initialize the clustering with the k-means algorithm on the couples $((x_1, y_1), \dots, (x_n, y_n))$. After that, we compute the linear regression estimators in each class. Then, we run 10 times the EM-algorithm, repeat this initialization 50 times, and keep the one which maximizes the log-likelihood function. Finally, to stop the algorithm, we could wait for any convergence, but the EM algorithm is known to check the convergence hypothesis, without converging, because of local maximum. Consequently, we choose to fix a minimum number of iterations to ensure non-local maximum, and to specify a maximum time of running. Between these two bounds, we stop if there is convergence of the log-likelihood and of the parameters (with a relative criteria), adapted from [21].

2.3. **Clustering with Gaussian mixture regression.** Suppose we know how many clusters there are, and assume that we get, from the observations, $\hat{\theta}$ such that $h_{\hat{\theta}}$ well approximate the unknown conditional density s. Then, we want to group the data into clusters between observations that seem similar. From a different point of view, we can look at this problem as a missing data problem. Indeed, the complete data are $((x_1, y_1, z_1), \ldots, (x_n, y_n, z_n))$ in which the latent variables are $Z = (Z_1, \ldots, Z_n)$ with $Z_i = (Z_{i,1}, \ldots, Z_{i,k})$ for $i \in \{1, \ldots, n\}$ is defined by

$$Z_{i,r} = \begin{cases} 1 & \text{if } Y_i \text{ arises from the } r^{th} \text{ subpopulation ;} \\ 0 & \text{otherwise.} \end{cases}$$

Thanks to the estimation $\hat{\theta}$, we could use the Maximum A Posteriori principle (later denoted MAP principle) to cluster data. Specifically, for all $i \in \{1, \ldots, n\}$, for all $r \in \{1, \ldots, k\}$, consider

$$\tau_{i,r}(\theta) = \frac{\pi_r \det(P_r) \exp\left(-\frac{1}{2}(P_r Y_i - \Phi_r X_i)^t (P_r Y_i - \Phi_r X_i)\right)}{\sum_{l=1}^{k} \pi_l \det(P_l) \exp\left(-\frac{1}{2}(P_l Y_i - \Phi_l X_i)^t (P_l Y_i - \Phi_l X_i)\right)}$$

the posterior probability of Y_i coming from the component number r. Then, the data are partitioned by the following rule:

$$Z_{i,r} = \begin{cases} 1 & \text{if } \tau_{i,r}(\hat{\theta}) > \tau_{i,l}(\hat{\theta}) \text{ for all } l \neq r \text{ ;} \\ 0 & \text{otherwise.} \end{cases}$$

2.4. **The model collection.** We want to deal with high-dimensional data, that is why we have to determine which variables are relevant for the Gaussian regression mixture clustering process. Indeed, we observe a small sample and we have to estimate a lot of coefficients: we have a problem of identifiability. The size n of the sample is smaller than $p \times m \times k$, the size of Φ. A way to solve this problem is to select a few variables to describe the problem. We then assume that we could estimate s by a sparse model. To reduce the dimension, we want to determine which variables are useful for the clustering, and which are not. This leads to the definition of an irrelevant variable.

Definition 2.1. *A variable indexed by $(j, z) \in [1, p] \times [1, m]$ is irrelevant for the clustering if $\Phi_{1,j,z} = \ldots = \Phi_{k,j,z} = 0$. We denote by J the relevant variables set.*

We denote by $x^{[J]}$ the restriction of x on J, and $\mathcal{H}_{(k,J)}$ the model with k components and with J for relevant variables set:

$$\mathcal{H}_{(k,J)} = \{y \in \mathbb{R}^m | x \in \mathbb{R}^p \mapsto h_\theta(y|x)\} ; \tag{7}$$

where

$$h_\theta(y|x) = \sum_{r=1}^{k} \frac{\pi_r \det(P_r)}{(2\pi)^{m/2}} \exp\left(-\frac{(P_r y - \Phi_r x^{[J]})^t (P_r y - \Phi_r x^{[J]})}{2}\right),$$

and

$$\theta = (\pi_1, \ldots, \pi_k, \Phi_1, \ldots, \Phi_k, \rho_1, \ldots, \rho_k) \in \Pi_k \times \left(\mathbb{R}^{|J|}\right)^k \times \left(\mathbb{R}_+^m\right)^k .$$

We will construct a model collection, by varying the number of components k and the relevant variables subset J.

3. Two procedures

The goal of our procedures is, given a sample $(x, y) = ((x_1, y_1), \ldots, (x_n, y_n))$, to discover the structure of the variables $Y|X$. Thus, we have to determine, according to the representation of $\mathcal{H}_{(k,J)}$, the number of clusters k, the relevant variables set J, and the parameters θ. To overcome this difficulty, we want to take advantage of the sparsity property of the ℓ_1-penalization to perform automatic variable selection in clustering for high-dimensional data. Then, we could compute another estimator restricted on active variables, which will work better because it is no longer an high-dimensional issue. Thus, we avoid shrinkage problems of the Lasso estimator. The first procedure takes advantage of the MLE, whereas the second one takes into account the matrix structure of Φ with a low rank estimation.

3.1. **Lasso-MLE procedure.** This procedure is decomposed into three main steps: we construct a model collection, then in each we compute the MLE, and we choose the best one among all the models. The first step consists of constructing a collection of models $\{\mathcal{H}_{(k,J)}\}_{(k,J)\in\mathcal{M}}$ in which $\mathcal{H}_{(k,J)}$ is defined by equation (7), and the model collection is indexed by $\mathcal{M} = K \times \mathcal{J}$. Denote $K \subset \mathbb{N}^*$ the possible number of components. We could bound K without loss of estimation. In practice, we could consider a mixture of 20 densities to approximate well a lot of density. Denote also $\mathcal{J}$ a collection of subsets of $\{1,\dots,p\} \times \{1,\dots,m\}$.
To detect the relevant variables, and construct the set $J \in \mathcal{J}$, we penalize the empirical contrast by an ℓ_1-penalty on the mean parameters proportional to $||\Phi_r||_1 = \sum_{j=1}^{p}\sum_{z=1}^{m}|\Phi_{r,j,z}|$. In the ℓ_1-procedures, the choice of the regularization parameters is often difficult: fixing the number of components $k \in K$, we propose to construct a data-driven grid G_k of regularization parameters by using the updating formulas of the mixture parameters in the EM algorithm. We can give a formula for λ, the regularization parameter, depending on which coefficient we want to cancel, for all $r \in [1,k], j \in [1,p], z \in [1,m]$:

$$\Phi_{r,j,z} = 0 \quad \Leftrightarrow \quad \lambda_{r,j,z} = \frac{|S_{r,j,z}|}{n\pi_r};$$

with $S_{r,j,z}$ defined in (5). It is a data-driven grid, depending on the observations. We could compute this from MLE estimations.
Then, for each $\lambda \in G_k$, we could compute the Lasso estimator defined by

$$\hat{\theta}^L_{(k,J)} = \underset{\theta\in\Theta}{\operatorname{argmin}} \left\{ \frac{1}{n}\sum_{i=1}^{n} \log(h_\theta(y_i|x_i)) + \lambda \sum_{r=1}^{k} \pi_r ||\Phi_r||_1 \right\}.$$

For a fixed number of mixture components $k \in K$ and a regularization parameter λ, we could use an EM algorithm, recalled in Appendix 8.1, to approximate this estimator. Then, for each $k \in K$, and for each $\lambda \in G_k$, we have constructed the relevant variables set J. We denote by $\mathcal{J}$ the collection of all these sets.
The second step consists of approximating the MLE

$$\hat{h}_{(k,J)} = \underset{t\in\mathcal{H}_{(k,J)}}{\operatorname{argmin}} \left\{ \frac{1}{n}\sum_{i=1}^{n} \log(t(y_i|x_i)) \right\};$$

using the EM algorithm for each model $(k,J) \in \mathcal{M}$.
The third step is devoted to model selection. We use the slope heuristic described in [3]. Explain briefly how it works. Firstly, models are grouping according to their dimension D, to obtain a model collection $\{\mathcal{H}_D\}_{D\in\mathcal{D}}$. The dimension of a model is the number of parameters estimated in the model. For each dimension D, let $\hat{h}_D$ be the estimator maximizing the likelihood among the estimators associated to a model of dimension D. Also, the function $D/n \mapsto \frac{1}{n}\sum_{i=1}^{n}\log(\hat{h}_D(y_i|x_i))$ has a linear behavior for large dimensions. We estimate the slope, denoted by $\hat{\kappa}$, which will be used to calibrate the penalty. The minimizer $\hat{D}$ of the penalized criterion $-\frac{1}{n}\sum_{i=1}^{n}\log(\hat{h}_D(y_i|x_i)) + 2\hat{\kappa}D/n$ is determined, and the estimator selected is $\hat{h}_{(k_{\hat{D}},J_{\hat{D}})}$.

3.2. **Lasso-rank procedure.** Whereas the previous procedure does not take into account the multivariate structure, we propose a second procedure to perform this point. For each model belonging to the collection $\mathcal{H}_{(k,J)}$, a subcollection is constructed, varying the rank of Φ. Let us describe this procedure.
As in the Lasso-MLE procedure, we first construct a collection of models, thanks to the ℓ_1-approach. We obtain an estimator for θ, denoted by $\hat{\theta}_{\text{Lasso}}$, for each model belonging to the collection. We could deduce the set of relevant variables, denoted by J, and this for all $k \in K$: we deduce $\mathcal{J}$ the collection of set of relevant variables.
The second step consists to construct a subcollection of models with rank sparsity, denoted by

$$\{\tilde{\mathcal{H}}_{(k,J,R)}\}_{(k,J,R)\in\tilde{\mathcal{M}}}$$

. The model $\{\tilde{\mathcal{H}}_{(k,J,R)}\}$ has k components, the set J for active variables, and R is the vector of the ranks of the matrix of regression coefficients in each group:

$$\tilde{\mathcal{H}}_{(k,J,R)} = \{y \in \mathbb{R}^m | x \in \mathbb{R}^p \mapsto h_\theta(y|x)\} \tag{8}$$

where

$$h_\theta(y|x) = \sum_{r=1}^{k} \frac{\pi_r \det(P_r)}{(2\pi)^{m/2}} \exp\left(-\frac{(P_r y - \Phi_r^{R(r)} x^{[J]})^t (P_r y - \Phi_r^{R(r)} x^{[J]})}{2} \right);$$

$$\theta = (\pi_1, \dots, \pi_k, \Phi_1^{R(1)}, \dots, \Phi_k^{R(k)}, \rho_1, \dots, \rho_k) \in \Pi_k \times \Psi_k^R \times \left(\mathbb{R}_+^m\right)^k;$$

$$\Psi_k^R = \left\{ (\Phi_1^{R(1)}, \dots, \Phi_k^{R(k)}) \in \left(\mathbb{R}^{|J|}\right)^k | \text{Rank}(\Phi_1) = R(1), \dots, \text{Rank}(\Phi_k) = R(k) \right\};$$

and $\tilde{\mathcal{M}} = K \times \mathcal{J} \times \mathcal{R}$. Denote $K \subset \mathbb{N}^*$ the possible number of components, $\mathcal{J}$ a collection of subsets of $\{1, \dots, p\} \times \{1, \dots, m\}$, and $\mathcal{R}$ the set of vectors of size $k \in K$ with ranks values for each mean matrix. We could compute the MLE under the rank constraint thanks to an EM algorithm. Indeed, we could constrain the estimation of Φ_r, for all r, to have a rank equal to $R(r)$, in keeping only the $R(r)$ largest singular values. More details are given in section 8.2. This leads to an estimator of the mean with row sparsity and low rank for each model. For the estimation of π and P, we keep those computed by the Lasso, denoted by $\hat{\pi}_{\text{Lasso}}$ and $\hat{P}_{\text{Lasso}}$.
As described in the above section, a model is selected using the slope heuristic.

4. Illustrative example

We illustrate our procedures on four different simulated datasets, adapted from [21], belonging to the model collection. They have been both implemented in Matlab, and the Matlab code is available from the author upon request. Firstly, we will present models used in these simulations. Then, we validate numerically each step, and we finally compare results of our procedures with others.

4.1. The model.

Let X be a sample of size n distributed according to multivariate standard Gaussian. We consider a mixture of two components, and we fix the dimension of the regressor and of the response variables to $p = m = 10$. Besides, we fix the number of active variables to 4 in each cluster. More precisely, the first four variables of Y are explained respectively by the four first variables of X. Fix $\pi = \left(\frac{1}{2}, \frac{1}{2}\right)$ and $P_r = I_m$ for all $r \in \{1, 2\}$.
The difficulty of the clustering is partially controlled by the signal-to-noise ratio. In this context, we could extend the natural idea of the SNR with the following definition, where $Tr(A)$ denotes the trace of the matrix A.

$$\text{SNR} = \frac{\text{Tr}(Var(Y))}{\text{Tr}(Var(Y|\beta_r = 0 \text{ for all } r \in \{1, \dots, k\}))}.$$

Remark that it only controls the distance between the signal with or without the noise, and not the distance between the both signals.
We compute four different models, varying n, the SNR, and the distance between the clusters. Details are available in the Table 1.

	Model 1	Model 2	Model 3	Model 4
n	2000	100	100	100
$\beta_{1\|J}$	3	3	3	5
$\beta_{2\|J}$	-2	-2	-2	3
σ	1	1	3	1
SNR	3.6	3.6	1.88	7.8

Table 1. Description of the different models.

Take a sample of $Y|X$ according to a Gaussian mixture, meaning in $\beta_r X$ and with covariance matrix $\Sigma_r = (P_r^t P_r)^{-1} = \sigma I_m$, for the cluster r. We run our procedures with the number of components varying in $\mathcal{K} = \{2, \dots, 5\}$.
The model 1 is chosen in the next section to illustrate well each step of the procedure (variable selection, models construction and model selection). It also illustrates that our procedures work of course in low dimension. Models 2, 3 and 4 are considered high-dimensional, because $m \times p \times k > n$. The model 2 is easier than the others, because clusters are not so close to each other according to the noise. Model 3 is constructed as the models 1 and 2, but n is small and the noise is more important. We will see that it gives difficulty for the clustering. Model 4 has a larger SNR, nevertheless, the problem of clustering is difficult, because each β_r is closer to the other rather than in the model 3.

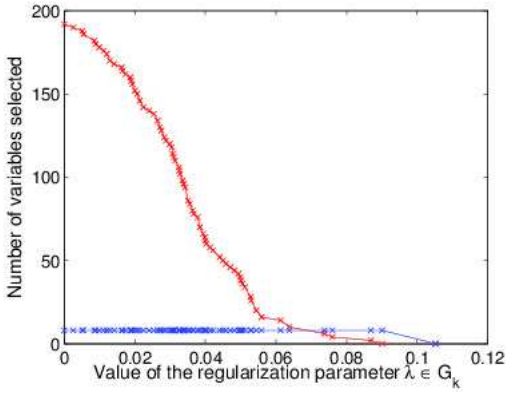

FIGURE 1. For one simulation, number of false relevant (in red color) and true relevant (in blue color) variables generated by the Lasso, by varying the regularization parameter λ in the grid G_2.

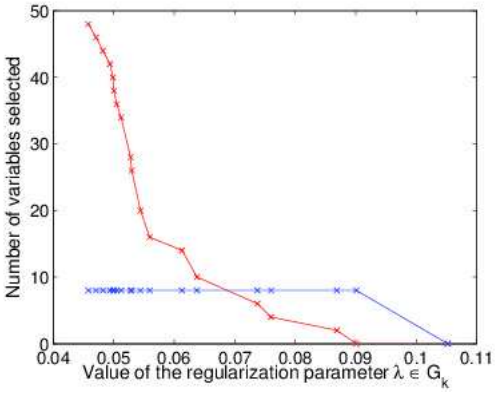

FIGURE 2. For one simulation, zoom in on number of false relevant (in red color) and true relevant (in blue color) variables generated by the Lasso, by varying the regularization parameter λ around the interesting values.

4.2. **Sparsity and model selection.** To illustrate the procedures, all the analyses made in this section are done from the model 1, since the choice of each step is clear.
Firstly, we compute the grid of regularization parameters. More precisely, each regularization parameter is computed from MLE estimations (using EM algorithm), and give an associated sparsity (computed by the Lasso estimator, using again the EM algorithm). In Figures (1) and (2), the collection of relevant variables selected by this grid are plotted.
Firstly, we could notice that the number of relevant variables selected by the Lasso decreases with the regularization parameter. We could analyze more precisely which variables are selected: if we select true relevant or false relevant variables. If the regularization parameter is not too large, the true active variables are selected. Even more, if the regularization parameter is well-chosen, we select only the true active variables. In our example, we remark that if $\lambda = 0.09$, we have selected exactly the true active variables. This grid construction seems to be well-chosen according to the simulations.
From this variable selection, each procedure (Lasso-MLE or Lasso-rank) leads to a model collection, varying the sparsity thanks to the regularization parameters grid.
The next step in the both procedures is the model selection. We have chosen the slope heuristic to select a model among the collection. Even if there is some theoretical results on this heuristic, nothing is done in this context. We recall here the main heuristic and confirm this in practice.
We want to select the best model by improving a penalized criterion. This penalty is computed by performing a linear regression on the couples of points
$\{(D/n; -\frac{1}{n}\sum_{i=1}^{n}\log(\hat{h}_D(y_i|x_i))\}$, D varying. This slope $\hat{\kappa}$ allows to have access to the best model, the one with dimension $\hat{D}$ minimizing $-\frac{1}{n}\sum_{i=1}^{n}\log(\hat{h}_D(y_i|x_i)) + 2\hat{\kappa}\frac{D}{n}$. In practice, we have to look if couples of points have a linear comportment . For each procedure, we construct the different model collection, and we have to justify this behavior. The graphs (3) and (4) represent the contrast γ_n (the log-likelihood) in function of the dimension of the models, for model collections constructed respectively by the Lasso-MLE procedure and by the Lasso-rank procedure. The couples are plotted by points, whereas the estimated slope is specified by a dotted line. We could observe more than a line (4 for the Lasso-MLE procedure, more for the Lasso-rank procedure). This phenomenon could be explained by a linear behavior for each mixture, fixing the number of classes, and the ranks. Nevertheless, slopes are almost the same, and select the same model. Note that we estimate the slope with the Capushe package [2].

4.3. **Assessment.** We compare our procedures to three other procedures on simulated models 1, 2, 3 and 4.

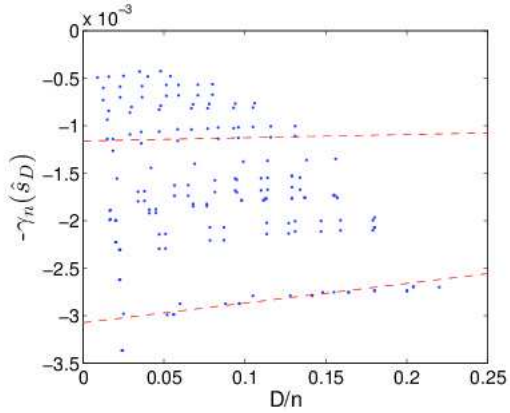

FIGURE 3. For one simulation, slope graph obtain by our Lasso-rank procedure. For large dimensions, we observe a linear part.

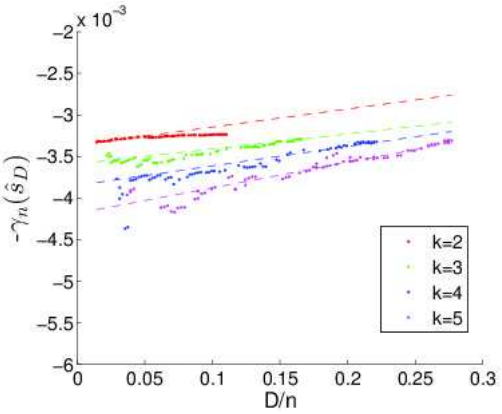

FIGURE 4. For one simulation, slope graph obtain by our Lasso-MLE procedure. For large dimensions, we observe a linear part.

Firstly, give some remarks about the model 1. For each procedure, we get good clustering and very low Kullback-Leibler divergence. Indeed, the sample size is large, and the estimations are good. That is the reason why we focus in this section on models 2, 3 and 4.
To compare our procedures with others, the Kullback-Leibler divergence with the true density and the ARI (the Adjusted Rand Index measures the similarity between two data clusterings, knowing that the closer to 1 the ARI, the more similar the two partitions) are computed, and we note which variables are selected, and how many classes are selected. For more details on the ARI, see [20].
From the Lasso-MLE model collection, we construct two models, to compare our procedures with. We compute the oracle (the model which minimizes the Kullback-Leibler divergence with the true density), and the model which is selected by the BIC criterion instead of the slope heuristic. Thanks to the oracle, we know how good we could get from this model collection for the Kullback-Leibler divergence, and how this model, as good it is possible for the contrast, performs the clustering.
The third procedure we compare with is the maximum likelihood estimator, assuming that we know how many clusters there are, fixed to 2. We use this procedure to show that variable selection is necessary.
In each case, we apply the MAP principle, to compare the clustering.
We do not plot the Kullback-Leibler divergence for the MLE procedure, because values are too high, and make the boxplots unreadable.
For the model 2, according to the Figure (5) for the Kullback-Leibler divergence, and Figure (6) for the ARI, the Kullback-Leibler divergence is small and the ARI is close to 1. The model collections are then well constructed. The model 3 is more difficult, because the noise is higher. That is why results, summarized in Figures (7) and (8), are not as good as the model 2. Nevertheless, our procedures lead to the best ARI, and the Kullback-Leibler divergences are close to the one of the oracle. We could make the same remarks for the model 4. In this study, the means are closer, according to the noise. Results are summarized in Figures (9) and (10). Note that the Kullback-Leibler divergence is smaller for the Lasso-MLE procedure, thanks to the maximum likelihood refitting. Moreover, the true model has not any matrix structure. If we look after the MLE, where we do not use the sparsity hypothesis, we could conclude that estimations are not satisfactory, which could be explained by an high-dimensional issue. The Lasso-MLE procedure, the Lasso-rank procedure and the BIC model work almost as well as the oracle, which mind that the models are well selected.

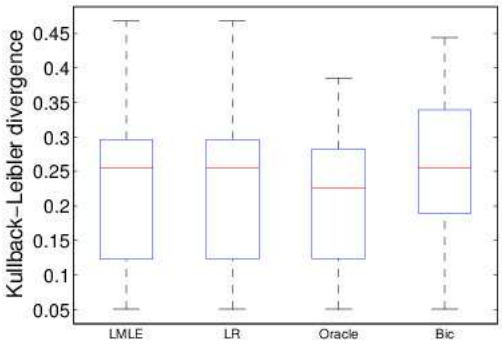

FIGURE 5. Boxplots of the Kullback divergence between the true model and the one selected by the procedure over the 20 simulations, for the Lasso-MLE procedure (LMLE), the Lasso-rank procedure (LR), the oracle (Oracle), the BIC estimator (BIC) for the model 2.

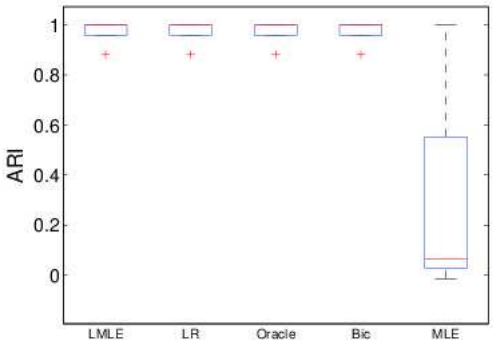

FIGURE 6. Boxplots of the ARI over the 20 simulations, for the Lasso-MLE procedure (LMLE), the Lasso-rank procedure (LR), the oracle (Oracle), the BIC estimator (BIC) and the MLE (MLE) for the model 2.

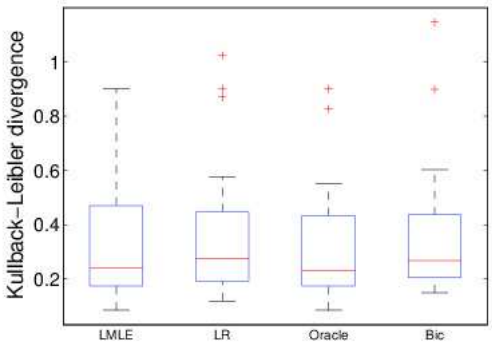

FIGURE 7. Boxplots of the Kullback divergence between the true model and the one selected by the procedure over the 20 simulations, for the Lasso-MLE procedure (LMLE), the Lasso-rank procedure (LR), the oracle (Oracle), the BIC estimator (BIC) for the model 3.

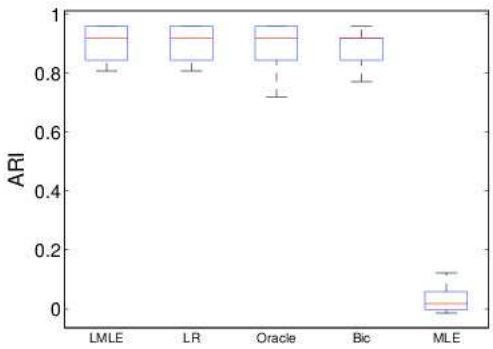

FIGURE 8. Boxplots of the ARI over the 20 simulations, for the Lasso-MLE procedure (LMLE), the Lasso-rank procedure (LR), the oracle (Oracle), the BIC estimator (BIC) and the MLE (MLE) for the model 3.

	Model 2		Model 3		Model 4	
Procedure	TR	FR	TR	FR	TR	FR
Lasso-MLE	8(0)	2.2(6.9)	8(0)	4.3(28.8)	8(0)	2(13, 5)
Lasso-rank	8(0)	24(0)	8(0)	24(0)	8(0)	24(0)
Oracle	8(0)	1.5(3.3)	7.8(0.2)	2.2(11.7)	8(0)	0.8(2.6)
BIC estimator	8(0)	2.6(15.8)	7.8(0.2)	5.7(64.8)	8(0)	2.6(11.8)

TABLE 2. Mean number {TR,FR} of true relevant and false relevant variables over the 20 simulations for each procedure, for the models 2, 3 and 4. The standard deviation are put into parenthesis.

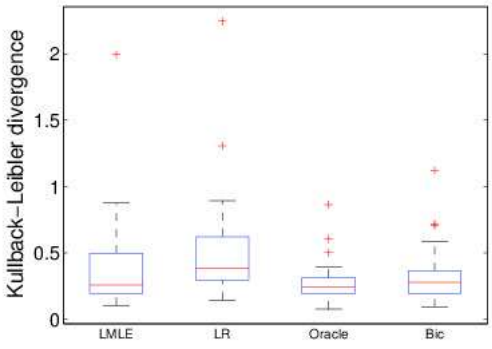

FIGURE 9. Boxplots of the Kullback divergence between the true model and the one selected by the procedure over the 20 simulations, for the Lasso-MLE procedure (LMLE), the Lasso-rank procedure (LR), the oracle (Oracle), the BIC estimator (BIC) for the model 4.

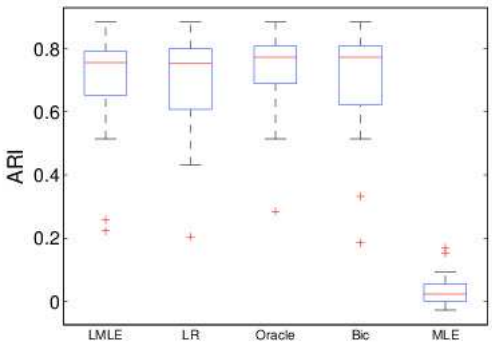

FIGURE 10. Boxplots of the ARI over the 20 simulations, for the Lasso-MLE procedure (LMLE), the Lasso-rank procedure (LR), the oracle (Oracle), the BIC estimator (BIC) and the MLE (MLE) for the model 4.

In Table (2), we summarize results about variable selection. For each model, for each procedure, we compute how many true relevant and the false relevant variables are selected.
The true model has 8 relevant variables, which is always recognized. The Lasso-MLE has less false relevant variables than the others, which means that the true structure was found. The Lasso-rank has 24 false relevant variables, because of the matrix structure: the true rank in each component was 4, then the estimator restricted on active variables is a 4×4 matrix, and we get 12 false relevant variables in each component. Nevertheless, we do not have more variables, that is to say the model constructed is the best possible. The BIC estimator and the oracle have a large variability for the false relevant variables.
For the number of components, we find that all the procedures have selected the true number 2.
Thanks to the MLE, the first procedure has good estimations (better than the second one). Nevertheless, depending on the data, the second procedure could be more attractive. If there is a matrix structure, for example if most of the variation of the response Y is caught by a small number of linear combinations of the predictors, the second procedure will work better.
We could conclude that the model collection is well constructed, and that the clustering is well-done.

5. Functional datasets

In different fields of applications, considering data are functions. The functional data analysis has been popularized first by Ramsay and Silverman in their book ([19]). It gives a description of main tools to analyze functional datasets. Another important book is the Ferraty and Vieu one's ([9]). However, the main part of the existing literature is concentrated on Y scalar and X functional. For example, we can cite Zhao et al. ([24]) for using wavelet basis in linear model, Yao et al. ([23]) for functional mixture regression, or Ciarleglio et al. ([6]) for using wavelet basis in functional mixture regression. In this section, we concentrate on Y and X both functional. Given a sample of curves, an important task is to search for homogeneous subgroups of curves using clustering. In a functional context, clustering leads to identify the individuals involved in the same or similar process, and is also useful to determine one representative curve per cluster from the noisy observations. We want to detect particular changes between several clusters. This implies a denoising and smoothing signal process so as to remove the noise and capture only the important patterns in the data. Here, we explain how our procedures can be applied in this context. Note that we could apply our procedure with scalar response and functional regressor, or, on the contrary, with functional response for scalar regressor. We explain how our procedure is generalized in the more difficult case, the other cases resulting of that.

5.1. **Functional regression model.** Suppose we observe a centered sample of functions $(x_i, y_i)_{i=1,\dots,n}$, associated with the variables (x, y), coming from a probability distribution with unknown conditional density s. We want to estimate this model by a functional mixture model: if the variables (x, y) come from the component r, there exists a function β_r such that

$$y(t) = \int_{I_x} x(s)\beta_r(s,t)ds + \epsilon(t), \tag{9}$$

where ϵ is a residual function. This linear model is introduced in Ramsay and Silverman's book [19]. They propose to project into basis and the response, and the regressors. We extend their model in mixture model, to consider several subgroups for a sample.
If we assume that, for all t, for all $i \in [1, n]$, $\epsilon_i(t) \sim N(0, \Sigma_r)$, the model (9) is an integrated version of the model (1). Depending on the class r, the linear reliance of y with respect to x is described by the function β_r.

5.2. **Projection into a wavelet basis.** To deal with functional data, we project them onto some basis, to obtain data as described in the Gaussian mixture regression models (1). In this paper, we choose to deal with wavelet basis, given that they represent localized features of functions in a sparse way. If the coefficients matrix X and Y are sparse, the regression matrix β has more chance to be sparse. Moreover, we could represent a signal with a few coefficients dataset, then it is a way to reduce the dimension. For details about the wavelet theory, see the Mallat's book [13].
Begin by an overview of some important aspects of wavelet basis.
Let ψ a real wavelet function, satisfying

$$\psi \in L^1 \cap L^2, t\psi \in L^1, \text{ and } \int_{\mathbb{R}} \psi(t)dt = 0.$$

We denote by ψ_{lh} the function defined from ψ by dyadic dilation and translation:

$$\psi_{lh}(t) = 2^{l/2}\psi(2^l t - h) \text{ for } (l, h) \in \mathbb{Z}^2.$$

We could define wavelet coefficients of a signal f by

$$d_{lh}(f) = \int_0^1 f(t)\psi_{lh}(t)dt \text{ for } (l, h) \in \mathbb{Z}^2.$$

Let φ be a scaling function related to ψ. We could define, for a signal f,

$$c_0(f) = \int_0^1 f(t)\varphi(t)dt.$$

The scaling function is chosen such that $\mathcal{B} = \{\varphi, \psi_{lh}\}_{l\geq 0, 0\leq h\leq 2^l-1}$ is an orthonormal basis of $L_2([0, 1])$. Note that scaling functions serve to construct approximations of the function of interest, while the wavelet functions serve to provide the details not captured by successive approximations.
Let $L \in \mathbb{N}^*$. For a signal f, we could define the approximation at the level L by

$$A_L = \sum_{l>L}\sum_{h\in\mathbb{Z}} d_{l,h}\psi_{l,h};$$

and f could be decomposed by the approximation at the level L and the details $(d_{l,h})_{l<L}$.
The decomposition of the basis between scaling function and wavelet function emphasizes on the local nature of the wavelets, and that is an important aspect in our procedures, because we want to know which details allow us to cluster two samples together. We could distinguish between two similar curves that only differ locally.
To deal with wavelet basis, assume that $p = 2^L$, $L \in \mathbb{N}^*$.
Consider the sample (x_i, y_i), and introduce the wavelet expansion of x_i in the basis $\mathcal{B}$: for all $t \in [0, 1]$,

$$x_i(t) = c_0(x_i)\varphi(t) + \sum_{l=0}^{+\infty}\sum_{h=0}^{2^l-1} d_{lh}(x_i)\psi_{lh}(t).$$

The collection $\{c_0(x_i), d_{lh}(x_i)\}_{l,h}$ is the Discrete Wavelet Transform of f in the basis $\mathcal{B}$.
Because we project onto an orthonormal basis, this leads to a n-sample $(X_1, \dots, X_n)$ of wavelet coefficient decomposition vectors, with

$$x_i = WX_i;$$

in which x_i is the vector of the discretized values of the signal, X_i the matrix of coefficients in the basis $\mathcal{B}$, and W a $p \times p$ matrix defined by φ and ψ. The DWT can be performed by a computationally

fast pyramid algorithm (see Mallat, [12]). In the same way, there exists $W^{'}$ such that $y_i = W^{'}Y_i$, with $Y = (Y_1, \ldots, Y_n)$ a n sample of wavelet coefficient decomposition vectors. Because the matrices W and $W^{'}$ are orthogonal, we keep the mixture structure, and the noise is also Gaussian. We could consider the wavelet coefficient dataset $(X, Y) = ((X_1, Y_1), \ldots, (X_n, Y_n))$, which is constituted of n observations whose probability distribution could be modeled by the finite Gaussian mixture regression model (1).
We could apply our procedures to this dataset, and obtain a clustering of the data. The notion of relevant variable is natural: the function φ or ψ_{lh} is irrelevant if it appears in none of the wavelet coefficient decomposition of the functions in each cluster.
An additional step could be done to reconstruct the curves for each cluster: knowing X, we could reconstruct the signal x from the coefficient dataset on a known basis.

5.3. **Numerical experiments.** We will illustrate our procedures on functional datasets by using the Matlab wavelet toolbox (see Misiti et al. in [16] for details). Firstly, we simulate functional datasets, where the true model belongs to the model collection. Then, we run our procedure on an electricity dataset, to cluster successive days. We have access to a time series, measured every half-hour, of a load consumption, on 70 days. We extract the signal of each day, and construct couples by each day and its eve, and we aim at clustering these couples. To finish, we test our procedures on the well-known Tecator dataset. This benchmark dataset corresponds to the spectrometric curves and fat contents of meat. These experiments illustrate different aspects of our procedures. Indeed, the simulated example proves that our procedures work in a functional context. The second example is a toy example used to validate the classification, on real data already studied, and in which we clearly understand the clusters. The last example illustrates the use of the classification to perform prediction, and the description given by our procedures to the model constructed.

5.3.1. *Simulated functional data.* Firstly, we simulate a mixture regression model. Let X be a sample of the noised cosinus function, discretized on a 15 points grid. Let Y be, depending on the class, either X, or the function $-X$, computed by a white-noise.
We use the Daubechies-2 basis at level 2 to decompose the signal on a wavelet basis.
Our procedures are run 20 times, and the number of classes are fixed to $\mathcal{K} = 2$. Then our procedures run on the projection are compared with the oracle among the collection constructed by the Lasso-MLE procedure, and with the model selected by the BIC criterion among this collection. The MLE is also computed.

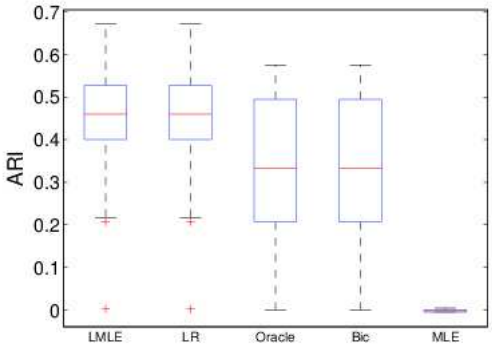

FIGURE 11. Boxplots of the ARI over the 20 simulations, for the Lasso-MLE procedure (LMLE), the Lasso-rank procedure (LR), the oracle (Oracle), the BIC estimator (BIC) and the MLE (MLE).

This simulated dataset proves that our procedures also perform clustering functional data, considering the projection dataset.

5.3.2. *Electricity dataset.* We also study the clustering on electricity dataset. This example is studied in [17]. We work on a sample of size 70 of couples of days, which is plotted in Figure 12. For each couple, we have access to the half-hour load consumption.

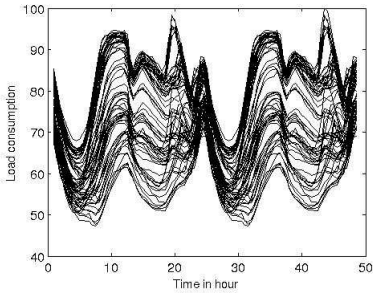

FIGURE 12. Plot of the 70-sample of half-hour load consumption, on the two days.

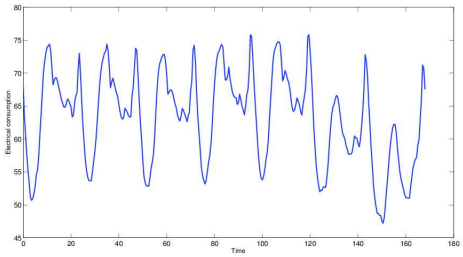

FIGURE 13. Plot of a week of load consumption.

As we said previously, we want to cluster couple of days. In Figure 13, we plot a week of consumption. A clustering for couple of days seems to be involved: weekdays together, and the others.
The regression model taken is X for the first day, and Y for the second day of each couple. Besides, discretization of each day on 48 points, every half-hour, is made available. In our opinion, a linear model is not appropriate, as the behavior from the eve to the day depends on which day we consider: there is a difference between working days and weekend days.
To apply our procedures, we project into wavelet basis X and Y. The symmlet-4 basis, at level 5 is used. We run our procedures with the number of clusters varying from 2 to 6. Our both procedures select a model with 4 components. The first one considers couples of weekdays, the second Friday-Saturday, the third component is Saturday-Sunday and the fourth considers Sunday-Monday. This result is faithful with the knowledge we have about these data. Indeed, working days have the same behavior, depending on the eve, whereas days off have not the same behavior, depending on working days, and conversely. Moreover, in the article [17], which also studied this example, they get the same classification.

5.3.3. *Tecator dataset.* This example deals with spectrometric data. More precisely, a food sample has been considered, which contained finely chopped pure meat with different fat contents. The data consist of a 100-channel spectrum of absorbances in the wavelength range $850 - 1050$ nm, and of the percentage of fat. We observe a sample of size 215. Those data have been studied in a lot of paper, cite for example Ferraty and Vieu's book [9]. They work on different approaches. They test prediction, and classification, supervised (where the fat content become a class, larger or smaller than 20%), or not (ignoring the response variable). In this work, we focus on clustering data according to the reliance between the fat content and the absorbance spectrum. We could not predict the response variable, because we do not know the class of a new observation. Estimate it is a difficult problem, in which we are not involved in this paper.
We will take advantage of our procedures to know which coefficient, in the wavelet basis decomposition of the spectrum, is useful to describe the fat content.
The sample will be split into two subsamples, 165 observations for the learning set, and 50 observations for the test set. We split it to have the same marginal distribution of the response in each sample.
The spectrum is a function, which we decompose into the Haar basis, at level 6. Nevertheless, our model did not take into account a constant coefficient to describe the response. Thereby, before run our procedure, we center and the Y according to the learning sample, and each function X_i for all observations in the whole sample. Then, we could estimate the mean of the response by the mean $\hat{\mu}$ over the learning sample.
We construct models on the training set by our procedure Lasso-MLE. Thanks to the estimations, we have access to active variables, and we could reconstruct signals keeping only active variables. We have also access to the a posteriori probability, which leads to know which observation is with high probability in which class. However, for some observations, the a posteriori probability do not ensure the clustering, being almost the same for different clusters. The procedure selects two models, which we describe here. In graphs (14) and (15), we represent clusters done on the training set for the different models.
The graph on the left is a candidate for representing each cluster, constructed by the mean of spectrum over an a posteriori probability greater than 0.6. We plot the curve reconstruction, keeping only active variables in the wavelet decomposition. On the right side, we present the boxplot of the fat values in each class, for observations with an a posteriori probability greater than 0.6.

The first model has two classes, which could be distinguish in the absorbance spectrum by the bump on wavelength around 940 nm. The first class is dominating, with $\hat{\pi}_1 = 0.95$. The fat content is smaller in the first class than in the second class. According to the signal reconstruction, we could see that almost all variables have been selected. This model seems consistent according to the classification goal.
The second model has 3 classes, and we could remark different important wavelength. Around 940 nm, there is some difference between classes, corresponding to the bump underline in the model 1, but also around 970 nm, with higher or smaller values. The first class is dominating, with $\hat{\pi}_1 = 0.89$. Just a few of variables have been selected, which give to this model the understanding property of which coefficient are discriminating.
We could discuss about those models. The first select only two classes, but almost all variables, whereas the second model has more classes, and less variables: there is a trade-off between clusters and variable selection for the dimension reduction.

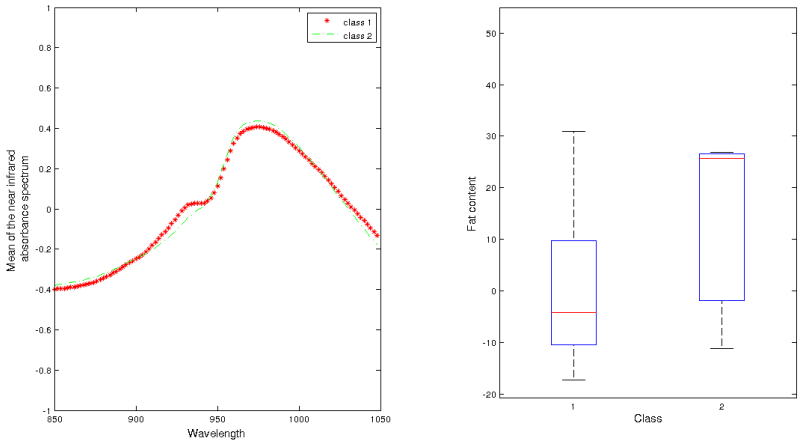

FIGURE 14. Summarized results for the model 1. The graph on the left is a candidate for representing each cluster, constructed by the mean of reconstructed spectrum over an a posteriori probability greater than 0.6. On the right side, we present the boxplot of the fat values in each class, for observations with an a posteriori probability greater than 0.6.

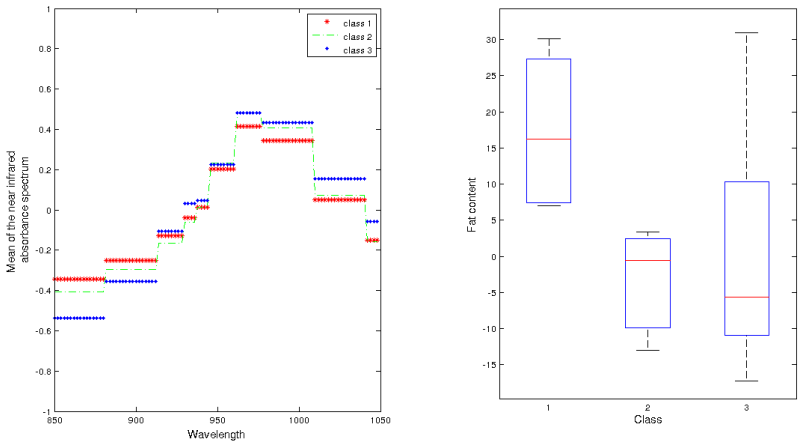

FIGURE 15. Summarized results for the model 2. The graph on the left is a candidate for representing each cluster, constructed by the mean of reconstructed spectrum over an a posteriori probability greater than 0.6. On the right side, we present the boxplot of the fat values in each class, for observations with an a posteriori probability greater than 0.6.

According to those classifications, we could compute the response according to the linear model. We use two ways to compute $\hat{y}$: either consider the linear model in the class selected by the MAP principle, or mix estimation in each class thanks to these a posteriori probability. We compute the Mean Absolute Percentage Error, $MAPE = \frac{1}{n}\sum_{i=1}^{n} |\frac{\hat{y}_i - y_i}{y_i}|$. Results are summarized in Table 3.

	Linear model in the class with higher probability	Mixing estimation
Model 1	0.20012	0.19765
Model 2	0.05515	0.05624

TABLE 3. Mean absolute percentage of error of the predicted value, for each model, for the learning sample.

Thus, we work on the test sample. We use the response and the regressors to know the a posteriori of each observation. Then, using our models, we could compute the predicted fat values from the spectrometric curve, as before according to two ways, mixing or choosing the classes.

	Linear model in the class with higher probability	Mixing estimation
Model 1	0.22196	0.21926
Model 2	0.20492	0.20662

TABLE 4. Mean absolute percentage of error of the predicted value, for each model, for the test sample.

Because the models are constructed on the learning sample, MAPE are lower than for the test sample. Nevertheless, results are similar, which saying that models are well constructed. This is particularly the case for the model 1, which is more consistent over a new sample.
To conclude this study, we could highlight the advantages of our procedure on these data. It provides a clustering of data, similar to the one done with supervised clustering in [9], but we could explain how this clustering is done.
This work has been done with the Lasso-MLE procedure. Nevertheless, the same kind of results have been get with the Lasso-rank procedure.

6. Conclusion

In this paper, two procedures are proposed to cluster regression data. Detecting the relevant clustering variables, they are especially designed for high-dimensional datasets. We use an ℓ_1-regularization procedure to select variables, and then deduce a reasonable random model collection. Thus, we recast estimations of parameters of these models into a general model selection problem. These procedures are compared with usual criteria on simulated data: the BIC criterion to select a model, or the cross-validation. We also compare to the maximum-likelihood estimator, and to the oracle when we know it. In addition, we compare our procedures to others on benchmark data.
From this work, several opportunities turn out to be interesting. In a work in progress, we will prove that the model selection is well done, in the both procedures. For the Lasso-MLE procedure, the theoretical result is available in [8].

7. Acknowledgment

I am indebted to Jean-Michel Poggi and Pascal Massart for suggesting me to study this problem, and for stimulating discussions. I am also grateful to Jean-Michel Poggi for carefully reading the manuscript and making many useful suggestions. I also thank Yves Misiti and Benjamin Auder for their help to speed up the code.

References

[1] T. W. Anderson. Estimating Linear Restrictions on Regression Coefficients for Multivariate Normal Distributions. *The Annals of Mathematical Statistics*, 22(3):327–351, 1951.
[2] J-P. Baudry, C. Maugis, and B. Michel. Slope heuristics: overview and implementation. *Statistics and Computing*, 22(1):455–470, 2012.

[3] L. Birgé and P. Massart. Minimal penalties for Gaussian model selection. *Probab. Theory Related Fields*, 138(1-2), 2007.
[4] P. Bühlmann and S. van de Geer. *Statistics for high-dimensional data*. Springer Series in Statistics. Springer, Heidelberg, 2011. Methods, theory and applications.
[5] F. Bunea, Y. She, and M. H. Wegkamp. Joint variable and rank selection for parsimonious estimation of high-dimensional matrices. *Ann. Statist.*, 40(5):2359–2388, 2012.
[6] A. Ciarleglio and R. T. Ogden. Wavelet-based scalar-on-function finite mixture regression models. Preprint, available at: arXiv:1312.0652, 2013.
[7] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. Discussion. *J. R. Stat. Soc., Ser. B*, 39:1–38, 1977.
[8] E. Devijver. Finite mixture regression: a sparse variable selection by model selection for clustering.
[9] F. Ferraty and P. Vieu. *Nonparametric functional data analysis : theory and practice*. Springer series in statistics. Springer, New York, 2006.
[10] C. Giraud. Low rank multivariate regression. *Electronic Journal of Statistics*, 5:775–799, 2011.
[11] A. J. Izenman. Reduced-rank regression for the multivariate linear model. *Journal of Multivariate Analysis*, 5(2):248–264, 1975.
[12] S. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:674–693, 1989.
[13] S. Mallat. *A wavelet tour of signal processing*. Academic Press, 1999.
[14] N. Meinshausen and P. Bühlmann. Stability selection. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 72(4):417–473, 2010.
[15] C. Meynet and C. Maugis-Rabusseau. A sparse variable selection procedure in model-based clustering. Rapport de recherche, September 2012.
[16] M. Misiti, Y. Misiti, G. Oppenheim, and J-M. Poggi. *Matlab Wavelet Toolbox User's Guide. Version 3*. The Mathworks, Inc., Natick, MA., July 2004.
[17] M. Misiti, Y. Misiti, G. Oppenheim, and J-M Poggi. Clustering signals using wavelets. *Lecture Notes in Computer Science*, 4507:514–521, 2007.
[18] T. Park and G. Casella. The Bayesian lasso. *J. Am. Stat. Assoc.*, 2008.
[19] J. O. Ramsay and B. W. Silverman. *Functional data analysis*. Springer series in statistics. Springer, New York, 2005.
[20] W. M. Rand. Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, 66, 1971.
[21] N. Stadler, P. Buhlmann, and S. Van de Geer. ℓ_1-penalization for mixture regression models. *Test*, 19(2):209–256, 2010.
[22] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc., Ser. B*, 58(1):267–288, 1996.
[23] F; Yao, Y. Fu, and T. C. M. Lee. Functional mixture regression. *Biostatistics*, 12(2):341–353, April 2011.
[24] Y. Zhao, R. T. Ogden, and P. T. Reiss. Wavelet-based lasso in functional linear regression. *Journal of Computational and Graphical Statistics*, 21(3):600–617, 2012.

8. Appendices

8.1. **EM algorithm for the Lasso estimator.** Introduced by Dempster et al. in [7], the EM (Expectation-Maximization) algorithm is used to compute maximum likelihood estimators, penalized or not.
The expected complete negative log-likelihood is denoted by

$$Q(\theta|\theta') = -\frac{1}{n} E_{\theta'}(l_c(\theta, Y, Z)|Y)$$

in which

$$\begin{aligned} l_c(\theta, Y, Z) = \sum_{i=1}^{n}\sum_{r=1}^{k} Z_{i,r} \log\left(\frac{\prod_{z=1}^{m}\rho_{r,z}}{(2\pi)^{m/2}} \exp\left(-\frac{1}{2}(P_r Y_i - X_i\Phi_r)^t(P_r Y_i - X_i\Phi_r)\right)\right) \\ + Z_{i,r}\log(\pi_r); \end{aligned}$$

with $Z_{i,r}$ are independent and identically distributed unobserved multinomial variables, showing the component-membership of the i^{th} observation in the finite mixture regression model.
The expected complete penalized negative log-likelihood is

$$Q_{\text{pen}}(\theta|\theta') = Q(\theta|\theta') + \lambda\sum_{r=1}^{k}\pi_r||\Phi_r||_1.$$

8.1.1. *Calculus for updating formula.*

- E-step: compute $Q(\theta|\theta^{(\text{ite})})$, or, equivalently, compute for $r = 1, \ldots, k$, $i = 1, \ldots, n$,

$$\begin{aligned}\hat{\gamma}_{i,r} &= E_{\theta^{(\text{ite})}}(Z_{i,r}|Y) \\ &= \frac{\pi_r^{(\text{ite})} \left(\prod_{z=1}^m \rho_{r,z}^{(\text{ite})}\right) \exp\left(-\frac{1}{2}\left(P_r^{(\text{ite})} Y_i - X_i \Phi_r^{(\text{ite})}\right)^t \left(P_r^{(\text{ite})} Y_i - X_i \Phi_r^{(\text{ite})}\right)\right)}{\sum_{l=1}^k \pi_l^{(\text{ite})} \left(\prod_{z=1}^m \rho_{r,z}^{(\text{ite})}\right) \exp\left(-\frac{1}{2}\left(P_r^{(\text{ite})} Y_i - X_i \Phi_l^{(\text{ite})}\right)^t \left(P_r^{(\text{ite})} Y_i - X_i \Phi_l^{(\text{ite})}\right)\right)}\end{aligned}$$

  This formula updates the clustering, thanks to the MAP principle.
- M-step: improve $Q_{\text{pen}}(\theta|\theta^{(\text{ite})})$.
  For this, rewrite the Karush-Kuhn-Tucker conditions. We have

$$\begin{aligned}&Q_{\text{pen}}(\theta|\theta^{(\text{ite})}) \\ &= -\frac{1}{n}\sum_{i=1}^n \sum_{r=1}^k E_{\theta^{(\text{ite})}}\left[Z_{i,r} \log\left(\frac{\prod_{z=1}^m \rho_{r,z}}{(2\pi)^{m/2}} \exp\left(-\frac{1}{2}(P_r Y_i - X_i \Phi_r)^t (P_r Y_i - X_i \Phi_r)\right)\right)|Y\right] \\ &- \frac{1}{n}\sum_{i=1}^n \sum_{r=1}^k E_{\theta^{(\text{ite})}}\left[Z_{i,r} \log \pi_r|Y\right] + \lambda \sum_{r=1}^k \pi_r ||\Phi_r||_1 \\ &= -\frac{1}{n}\sum_{i=1}^n \sum_{r=1}^k -\frac{1}{2}(P_r Y_i - X_i \Phi_r)^t (P_r Y_i - X_i \Phi_r) E_{\theta^{(\text{ite})}}\left[Z_{i,r}|Y\right] \\ &- \frac{1}{n}\sum_{i=1}^n \sum_{r=1}^k \sum_{z=1}^m \log\left(\frac{\rho_{r,z}}{\sqrt{2\pi}}\right) E_{\theta^{(\text{ite})}}\left[Z_{i,r}|Y\right] \qquad (10) \\ &- \frac{1}{n}\sum_{i=1}^n \sum_{r=1}^k E_{\theta^{(\text{ite})}}\left[Z_{i,r}|Y\right] \log \pi_r + \lambda \sum_{r=1}^k \pi_r ||\Phi_r||_1. \qquad (11)\end{aligned}$$

  Firstly, we optimize this formula with respect to π: it is as the optimization of

$$-\frac{1}{n}\sum_{i=1}^n \sum_{r=1}^k \hat{\gamma}_{i,r} \log(\pi_r) + \lambda \sum_{r=1}^k \pi_r ||\Phi_r||_1.$$

  We obtain

$$\pi_r^{(\text{ite}+1)} = \pi_r^{(\text{ite})} + t^{(\text{ite})}\left(\frac{\sum_{i=1}^n \hat{\gamma}_{i,r}}{n} - \pi_r^{(\text{ite})}\right);$$

  with $t^{(\text{ite})} \in (0, 1]$, the largest value in the grid $\{\delta^k, k \in \mathbb{N}\}$, with $0 < \delta < 1$, such that the function is not increasing.

  To optimize (10) with respect to (Φ, ρ), we could rewrite the expression: it is similar to the optimization of

$$-\frac{1}{n}\sum_{i=1}^n \left(\hat{\gamma}_{i,r} \sum_{z=1}^m \log(\rho_{r,z}) - \frac{1}{2}(P_r \tilde{Y}_i - \tilde{X}_i \Phi_r)^t (P_r \tilde{Y}_i - \tilde{X}_i \Phi_r)\right) + \lambda \pi_r ||\Phi_r||_1$$

  for all $r = 1, \ldots, k$, which is equivalent to the optimization of

$$-\frac{1}{n} n_r \sum_{z=1}^m \log(\rho_{r,z}) + \frac{1}{2n}\sum_{i=1}^n \sum_{z=1}^m \left(\rho_{r,z} \tilde{Y}_{i,r,z} - \Phi_{r,z} \tilde{X}_{i,r}\right)^2 + \lambda \pi_r ||\Phi_r||_1;$$

  where $n_r = \sum_{i=1}^n \hat{\gamma}_{i,r}$. The minimum in $\rho_{r,z}$ is the function which cancel its partial derivative with respect to $\rho_{r,z}$:

$$-\frac{n_r}{n}\frac{1}{\rho_{r,z}} + \frac{1}{2n}\sum_{i=1}^n 2\tilde{Y}_{i,r,z}\left(\rho_{r,z}\tilde{Y}_{i,r,z} - \Phi_{r,z}\tilde{X}_{i,r}\right) = 0$$

  for all $r \in [1, k]$, for all $z \in [1, m]$, which is equivalent to

$$-1+\frac{1}{n_r}\rho_{r,z}^2\sum_{i=1}^{n}\tilde{Y}_{i,z}^2-\frac{1}{n_r}\rho_{r,z}\sum_{i=1}^{n}\tilde{Y}_{i,z}\Phi_{r,z}\tilde{X}_{i,r}=0$$

$$\Leftrightarrow -1+\rho_{r,z}^2\frac{1}{n_r}||\tilde{Y}_z||_2^2-\rho_{r,z}\frac{1}{n_r}\langle\tilde{Y}_{i,z,r},\Phi_{r,z}\tilde{X}_{i,r}\rangle=0.$$

The discriminant is

$$\Delta=\left(-\frac{1}{n_r}\langle\tilde{Y}_{i,z,r},\Phi_{r,z}\tilde{X}_{i,r}\rangle\right)^2-\frac{4}{n_r}||\tilde{Y}_z||_2^2.$$

Then, for all $r\in\{1,\ldots,k\}$, for all $z\in\{1,\ldots,m\}$,

$$\rho_{r,z}=\frac{n_r\langle\tilde{Y}_{i,z,r},\Phi_{r,z}\tilde{X}_{i,r}\rangle+\sqrt{\Delta}}{2n_r||\tilde{Y}_z||_2^2}.$$

We could also look at the equation (10) as a function of the variable Φ: according to the partial derivative with respect to $\Phi_{r,j,z}$, we obtain for all $z\in\{1,\ldots,m\}$, for all $r\in\{1,\ldots,k\}$, for all $j\in\{1,\ldots,p\}$,

$$\sum_{i=1}^{n}\tilde{X}_{i,r,j}\left(\rho_{r,z}\tilde{Y}_{i,r,z}-\sum_{j_2=1}^{p}\tilde{X}_{i,r,j_2}\Phi_{r,j_2,z}\right)-n\lambda\pi_r\text{sgn}(\Phi_{r,j,z})=0.$$

Then, for all $r\in\{1,\ldots,k\}, j\in\{1,\ldots,p\}, z\in\{1,\ldots,m\}$,

$$\Phi_{r,j,z}=\frac{\sum_{i=1}^{n}\tilde{X}_{i,r,j}\rho_{r,z}\tilde{Y}_{i,r,z}-\sum_{\substack{j_2=1\\j_2\neq j}}^{p}\tilde{X}_{i,r,j}\tilde{X}_{i,r,j_2}\Phi_{r,j_2,z}-n\lambda\pi_r\text{sgn}(\Phi_{r,j,z})}{||\tilde{X}_{r,j}||_2^2}.$$

To reduce notations, let, for all $r\in\{1,\ldots,k\}, j\in\{1,\ldots,p\}, z\in\{1,\ldots,m\}$,

$$S_{r,j,z}=-\sum_{i=1}^{n}\tilde{X}_{i,r,j}\rho_{r,z}\tilde{Y}_{i,r,z}+\sum_{\substack{j_2=1\\j_2\neq j}}^{p}\tilde{X}_{i,r,j}\tilde{X}_{i,r,j_2}\Phi_{r,j_2,z}.$$

Then

$$\begin{aligned}\Phi_{r,j,z}&=\frac{-S_{r,j,z}-n\lambda\pi_r\text{sgn}(\Phi_{r,j,z})}{||\tilde{X}_{r,j}||_2^2}\\&=\begin{cases}\frac{-S_{r,j,z}+n\lambda\pi_r}{||\tilde{X}_{r,j}||_2^2} & \text{if } S_{r,j,z}>n\lambda\pi_r\\ -\frac{S_{r,j,z}+n\lambda\pi_r}{||\tilde{X}_{r,j}||_2^2} & \text{if } S_{r,j,z}<-n\lambda\pi_r\\ 0 & \text{else.}\end{cases}\end{aligned}$$

From these equalities, we could write the updating formulae:

$$\begin{aligned}\pi_r^{(\text{ite}+1)}&=\pi_r^{(\text{ite})}+t^{(\text{ite})}\left(\frac{\sum_{i=1}^{n}\hat{\gamma}_{i,r}}{n}-\pi_r^{(\text{ite})}\right);\\ \rho^{(\text{ite}+1)}&=\frac{\langle\tilde{Y}_{i,z,r},\Phi_{r,z}^{(\text{ite})}\tilde{X}_{i,r}\rangle+n_r\sqrt{\Delta}}{2||\tilde{Y}_z||_2^2};\\ \Phi_{r,j,z}^{(\text{ite}+1)}&=\begin{cases}\frac{-S_{r,j,z}^{(\text{ite})}+n\lambda(\pi_r^{(\text{ite})})}{||\tilde{X}_{r,j}||_2^2} & \text{if } S_{r,j,z}^{(\text{ite})}>n\lambda(\pi_r^{(\text{ite})});\\ -\frac{S_{r,j,z}^{(\text{ite})}+n\lambda(\pi_r^{(\text{ite})})}{||\tilde{X}_{r,j}||_2^2} & \text{if } S_{r,j,z}^{(\text{ite})}<-n\lambda(\pi_r^{(\text{ite})});\\ 0 & \text{else.}\end{cases}\end{aligned}$$

with, for $j\in\{1,\ldots,p\}, r\in\{1,\ldots,k\}, z\in\{1,\ldots,m\}$,

$$\begin{aligned}&S_{r,j,z}^{(\text{ite})}=-\sum_{i=1}^{n}\tilde{X}_{i,r,j}\rho_z^{(\text{ite})}\tilde{Y}_{i,r,z}+\sum_{j_2=1,j_2\neq j}^{p}\tilde{X}_{i,r,j}\tilde{X}_{i,r,j_2}\Phi_{r,j_2,z}^{(\text{ite})};\\ &n_r=\sum_{i=1}^{n}\hat{\gamma}_{i,r};\\ &(\tilde{Y}_{i,r},\tilde{X}_{i,r})=\sqrt{\hat{\gamma}_{i,r}}(Y_i,X_i).\end{aligned}$$

and $t^{(m)} \in (0,1]$, the largest value in the grid $\{\delta^k, k \in \mathbb{N}\}$, $0 < \delta < 1$, such that the function is not increasing.

8.2. **EM algorithm for the rank procedure.** To take into account the matrix structure, we want to make a dimension reduction on the rank of the mean matrix. If we have known to which cluster each sample belonged, we could do analysis of linear model in each component.
Indeed, an estimator of fixed rank is known in the linear regression case: denoting A^+ the Moore-Penrose pseudo-inverse of A, and $[A]_r = UD_rV^t$ in which D_r is obtained from D by setting $(D_q)_{i,i} = 0$ for $i \geq q+1$, with UDV^t the singular decomposition of A, if $Y = \beta X + \Sigma$, an estimator of β with rank r is $\hat{\beta}_r = [((X^tX)^+X^tY]_r$.
We do not know the clustering of the sample, but the E-step in the EM algorithm compute this.
We suppose in this case that Σ_r and π_r are known, for all $r \in \{1, \ldots, k\}$. We use this algorithm to determine Φ_r, for all $r \in \{1, \ldots, k\}$, with ranks fixed to $R(1), \ldots, R(k)$.

- E-step: compute for $r = 1, \ldots, k$, $i = 1, \ldots, n$,

$$\begin{aligned}\hat{\gamma}_{i,r} &= E_{\theta^{(\text{ite})}}(Z_{i,r}|Y) \\ &= \frac{\pi_r^{(\text{ite})} \left(\prod_{z=1}^m \rho_{r,z}^{(\text{ite})}\right) \exp\left(-\frac{1}{2}\left(P^{(\text{ite})}Y_i - X_i\Phi_r^{(\text{ite})}\right)^t \left(P^{(\text{ite})}Y_i - X_i\Phi_r^{(\text{ite})}\right)\right)}{\sum_{l=1}^k \pi_l^{(\text{ite})} \left(\prod_{z=1}^m \rho_{r,z}^{(\text{ite})}\right) \exp\left(-\frac{1}{2}\left(P^{(\text{ite})}Y_i - X_i\Phi_l^{(\text{ite})}\right)^t \left(P^{(\text{ite})}Y_i - X_i\Phi_l^{(\text{ite})}\right)\right)}\end{aligned}$$

- M-step: assign each observation in its estimated cluster, by the MAP principle applied thanks to the E-step. We say that Y_i comes from component number $\underset{r\in\{1,\ldots,k\}}{\text{argmax}}\, \hat{\gamma}_{i,r}$. Then, we can define $\tilde{\beta}_r^{(\text{ite})} = (X_{|r}^t X_{|r})^{-1} X_{|r}^t Y_{|r}$, in which $X_{|r}$ and $Y_{|r}$ are a restriction of the sample to the cluster r, which we decompose in singular value with $\tilde{\beta}_r^{(\text{ite})} = USV^t$. Using the singular value decomposition described before, we obtain the estimator.

Inria Select, Université Paris Sud, Bât. 425, 91405 Orsay Cedex, France
E-mail address: emilie.devijver@math.u-psud.fr